Description

COMPUTER CODE INTRUSION DETECTION SYSTEM BASED ON ACCEPTABLE RETRIEVALS

Inventor: Carey Nachenberg

Related Application

This patent application claims priority upon commonly assigned U.S. patent application serial no. 10/612,198 filed July 1, 2003, entitled "Real-Time Training for a Computer Code Intrusion Detection System", which patent application is hereby incorporated by reference in its entirety into the present patent application.

Technical Field

This invention pertains to the field of thwarting intrusions perpetrated by malicious attackers to computer code (e.g., databases).

Background Art

The background art includes intrusion thwarting systems where the computer code being attacked is a database. Such systems are called database intrusion detection systems. Some of these systems utilize offline non-real-time training in order to detect suspicious or anomalous activity. Examples of offline non-real-time database intrusion detection systems are described in *Lee, et al.*, "Learning Fingerprints for a Database Intrusion Detection System", ESORICS 2002, pp. 264-279, published in

November 2002 by Springer-Verlag, Berlin and Heidelberg, Germany; and *C. Chung*, et al., "DEMIDS: A Misuse Detection System for Database Systems", Department of Computer Science, University of California at Davis, Davis, California, October 1, 1999.

A common flaw in database intrusion detection systems of the prior art is that such systems fail to protect the database against insider attempts to steal large amounts of data using legitimate business processes. For example, such a system may allow a given service representative to access fields and tables within the database containing customer credit card information. Normally, a representative might access 5 to 10 accounts per hour in order to service customers. That is fine until the customer service representative decides to launch an insider attack on the database, procuring large amounts of consumer credit card information, which he then uses for nefarious purposes. The present invention is designed to protect against that and other attacks.

Disclosure of Invention

Methods, apparati, and computer-readable media for protecting computer code (1) from malicious retrievers (3). A method embodiment of the present invention comprises the steps of generating (22) retrieval information characteristic of data sent to a retriever (3) by the computer code (1) in response to a retrieval command (5) issued by the retriever (3); accessing at

least one rule (6) using at least some of said retrieval information as an input to said at least one rule (6); and, when said at least one rule (6) informs that the retrieval is not acceptable, flagging (28) the retrieval command (5) as suspicious.

Brief Description of the Drawings

These and other more detailed and specific objects and features of the present invention are more fully disclosed in the following specification, reference being had to the accompanying drawings, in which:

Figure 1 is a block diagram illustrating embodiments of the present invention.

Figure 2 is a flow diagram illustrating an operational phase of the present invention.

Figure 3 is a flow diagram illustrating a training phase of the present invention.

Figure 4 is a flow diagram illustrating a system administrator phase of the present invention.

Figure 5 is a diagram illustrating typical contents within state table 18 of the present invention.

Figure 6 is a diagram illustrating typical contents within rule table 6 of the present invention.

Detailed Description of the Preferred Embodiments

This invention has applicability to any code intrusion detection system, i.e., any system in which computer code 1 is susceptible to being attacked by commands 5 which may be malicious, due to malicious intent on the part of the user 3 who issues the command 5. As used herein, "user" can refer to a client computer 3 and/or to a human who has control of computer 3. As illustrated in Figure 1, there can be a plurality N of users 3, where N is any positive integer. "User" is sometimes referred to herein as "retriever".

Most of the following description illustrates the special case where the computer code 1 is a database 1. Database 1 can be any type of database, such as a relational database or a flat file. When database 1 is a relational database, commands 5 are typically written in a SQL language. As used herein, "SQL" is taken in the broad sense to mean the original language known as SQL (Structured Query Language), any derivative thereof, or any structured query language used for accessing a relational database. In the case where computer code 1 is not a relational database, the commands can be written in another language, such as XML. Database 1 may have associated therewith an internal audit table 11 and/or an external database log file 12 for storing audit and/or ancillary information pertaining to database 1. Database 1 is typically packaged within a dedicated computer

known as a database server 2, which may also contain database communications module 15 and other modules not illustrated.

Computer code intrusion detection system (IDS) 19 (and its special case, database intrusion detection system 19) encompasses modules 4, 6-9, 13, 17, and 18. Modules 1, 4, 6-9, 11-13, 15, 17, and 18 can be implemented in software, firmware, hardware, or any combination thereof, and are typically implemented in software. Figure 1 illustrates the case where modules 4, 6-9, 13, 17, and 18 are stand-alone modules separate from database server 2. However, these modules could just as well be incorporated within database server 2, e.g., they could be incorporated within database communications module 15. Thus, intrusion detection system 19 could be published by a third party as a standalone package on any type of computer-readable medium, or bundled by the manufacturer of the database 1 with module 15. The purpose of intrusion detection system 19 is to protect computer code 1 from users 3 that have nefarious intent. For example, such users may desire to steal (possibly large amounts of) credit card information from database 1.

One method embodiment of the present invention comprises three phases: a training phase, a system administrator phase, and an operational phase. Figure 2 illustrates the operational phase of the present invention. At optional step 20, computation module 7 extracts an input vector from a retrieval command 5,

using any technique of real-time auditing and/or in-line interception described below in conjunction with step 32. The extraction is typically done in real time or quasi-real-time. As used herein, "real time" means "during a short time interval surrounding the event". Thus, observing a command 5 in real time means that the command 5 is observed during a short time interval surrounding the instant that the command 5 enters the database 1.

A retrieval command 5 is any command by which a retrieving user 3 seeks to retrieve information from the database 1. The input vector characterizes the retrieval command 5 and comprises at least one parameter from the group of parameters comprising: canonicalized commands; the dates and times at which the commands 5 access the computer code 1; logins (user IDs, passwords, catch phrases, etc.) of users 3 issuing the commands 5; the identities of users 3 issuing the commands 5; the departments of the enterprise in which the users 3 work, or other groups to which the users 3 belong; the applications (i.e., software programs or types of software programs) that issue the commands 5; the IP addresses of the issuing computers 3; identities of users 3 accessing a given field or fields within the computer code 1; the times of day that a given user 3 accesses a given field or fields within the computer code 1; the fields or combination of fields being accessed by given commands 5; and tables or combinations of tables within the computer code 1 accessed by the commands.

A canonicalized command is a command 5 stripped of its literal field data. Literal field data is defined as a specific value of a parameter. Thus, for example, let us assume that the command 5 is:

SELECT NAME FROM PATIENTS WHERE NAME LIKE 'FRANK' AND AGE > 25

In this case, the literal field data is "FRANK" and "25". Thus, a canonicalized form of the command 5 is:

SELECT NAME FROM PATIENTS WHERE NAME LIKE * AND AGE > *

Literal fields can include literal numbers (plain numbers), dates, times, strings, and potentially named ordinal values (symbolic words used to represent numbers, e.g., "January" represents the first month, "Finance" represents department 54, etc.).

In one embodiment, a retrieval command 5 is subjected to step 20 only if the fields mentioned in the command 5 appear on a preselected list of fields deemed to be important, e.g., credit card and password fields. In other embodiments, the operational phase is performed without the need to extract an input vector, and thus step 20 is not performed at all.

At step 21, the retrieval command 5 is forwarded to the database 1 for processing. When the database 1 finishes processing the retrieval command 5, it normally sends back to user 3 the requested data in the form of rows plus columns and/or tables. A single row of data may contain a credit card number,

expiration date, and customer name, i.e., three columns worth of data. A second row of data then would contain a second credit card number, a second expiration date, and a second customer name.

At step 22, computation module 7 observes this response by database 1 (using any technique of real-time auditing and/or in-line interception described below in conjunction with step 32); and generates retrieval information therefrom. This retrieval information is optionally stored in state table 18, potentially along with one or more pieces of information from the input vector (e.g., to maintain data such as "users of the SUPPORT group retrieved an average of 10 customer records per hour"). State table 18 can maintain statistics on client 3 access to particular fields, associating the client 3 with the types of data that the client 3 is accessing. Clients 3 can be identified by user-ID ("Carey"), group-membership ("Average statistics for all members of the FINANCE group"), group-ID ("FINANCE group"), as well as potentially source IP address, machine name identification, client application, or other combinations of zero or more elements of the input vector. State table 18 stores a set of statistics associated with one or more of these client 3 identifiers. State table 18 may also group its data based on other attributes in the input vector, including the set of referenced fields, etc. (see point 8 below). For example:

CAREY's statistics:

1. has downloaded 2000 credit card rows total
2. downloads credit card rows at a rate of 10 per hour during business hours
3. downloads credit card rows at a rate of 3 per hour during off hours
4. has downloaded 1500 password rows total
5. downloads password rows at a rate of 10 per hour during business hours
6. downloads password rows at a rate of 3 per hour during off hours
7. downloads password rows at an average rate of 3 per request
8. For commands that attempt to access fields {USER, PASSWORD, SSN}, the average number of retrieved rows is 1.
9. etc…

FINANCE's average user statistics:

1. has downloaded 23000 credit card rows total
2. average finance user downloads credit card rows at a rate of 7 per hour during business hours
3. downloads credit card rows at a rate of 1 per hour during off hours
4. etc…

statistics for computer at IP Address 1.2.3.4:

etc.

etc.

etc.

The statistics can be maintained for only those fields deemed critical by the database administrator 10, or for all fields accessed. Clearly, many types of statistics can be maintained, including:

1. average number of row retrievals per given time unit (minutes, hours, seconds)
2. standard deviation of row retrievals per given time unit
3. average number of columns retrieved per time unit, etc.

Typical contents of a state table 18 having three entries are illustrated in Figure 5. In the first entry, an input vector was not calculated (at step 20), because here the operational phase is operating on a command by command basis. Thus, there is no need to track any identifying information for a particular command 5, because it is the present command 5 that is being processed.

"Retrieval information" consists of two components: one or more retrieval vectors, and statistical information. As used herein, "retrieval vector" comprises at least one of the following: the number of rows retrieved; the number of columns

retrieved; the number of tables retrieved; an identification of the columns that were retrieved; and an identification of the tables that were retrieved. Thus, in the present example of entry 1, the retrieval vector can be represented as [5 rows; 3 columns; columns A, J, and K]. As used herein, "statistical information" means any statistics that can be generated from the retrieval, either in conjunction with data stored in state table 18, or on its own. Thus, "statistical information" can comprise one or more of the following statistics: the rate of retrieving rows; the rate of retrieving columns; the rate of retrieving tables; the average number of rows retrieved per retrieval command 5 for a given input vector (or subset of an input vector); the average number of columns retrieved per retrieval command 5 for a given input vector; the average number of tables retrieved per retrieval command 5 for a given input vector; the percentage of retrieval commands 5 for which a given column is accessed; the percentage of retrieval commands 5 for which a given table is accessed; the percentage of retrieval commands 5 for which a given combination of columns is accessed; and the percentage of retrieval commands 5 for which a given combination of tables is accessed.

Note that some of these statistics are compilable across many commands 5, and some are compilable within a single command 5. In the present example of entry 1 in Figure 5, there are two

pieces of statistical information that have been generated by computation module 7 as a result of this particular command 5 accessing this particular database 1: S, the number of rows per second that are retrieved; and D, the number of columns per second that are retrieved. In this example, S=2000 rows per second and D=2300 columns per second.

At step 23, computation module 7 uses retrieval information to access at least one rule 6 pertaining to retrievals. The rules 6 can define acceptable and/or unacceptable retrievals, and can be stored in any manner known to one of ordinary skill in the art. In one embodiment, at least one rule 6 comprises a pre-established table containing rules for acceptable and/or unacceptable retrievals as illustrated in Figure 6. In the illustrated example, rule table 6 has four entries. In the first entry, there is no input vector, since the corresponding rule is independent of any particular input vector. (It may be said that the input vector is wildcarded.) This emphasizes the fact that it is not necessary for table 6 to be accessed (indexed) by an input vector. In this example, the cognizant rule, rule 5, states: "no more than 1000 rows per second can ever be retrieved by anybody".

At step 26, computation module 7 determines whether table 6 indicates that the retrieval is acceptable or unacceptable. The matching of the retrieval information from table 18 to the rule

in table 6 can be performed by any technique known to those of ordinary skill in the art. If table 6 indicates that the retrieval is acceptable, the retrieval is allowed to proceed at step 27, i.e., the requested data is sent to the requesting user 3.

If, on the other hand, the retrieval information from table 18 does not satisfy the corresponding rule in table 6, module 8 flags the current command 5 as being suspicious at step 28. Then a post-flagging protocol is performed by module 9 at step 29. In the illustrated example, the retrieval information "S=2000 rows per second" violates the rule "no more than 1000 rows per second can ever be retrieved by anybody". Thus, steps 28 and 29 are executed.

Execution of the post-flagging protocol at step 29 entails execution of at least one of the following steps: an alert is sent to the system administrator 10; an audit log is updated; the command 5 is not allowed to access the computer code 1; the command 5 is allowed to access the computer code 1, but the access is limited in some way (for example, the amount of data sent back to user 3 is limited); the command 5 is augmented, e.g., investigational code is inserted into the command 5 to provoke an audit trail; the user 3 sending the command 5 is investigated. The latter investigation can be performed by computer means (e.g., sending out a digital trace to determine

the identity of the user 3) and/or by off-line means (sending a human private investigator to spy on user 3).

The above example illustrates an embodiment in which table 6 is accessed by retrieval information but not by an input vector. In other embodiments, an input vector (or more than one input vector, as long as the input vectors are from the same command 5), in addition to retrieval information, is used to access table 6. For example, consider the second entry illustrated in Figure 6. The four rules set forth in said entry 2 are associated with a particular input vector $L_1F_1A_1$. These rules, which are more fully described below in conjunction with the training phase, are valid only with respect to specific input vector $L_1F_1A_1$.

The above examples illustrate the case where the operational phase is performed on a command by command basis. In other embodiments, the retrieval information can be compiled on other bases, for example, with respect to all commands 5 that are executed during a given time period that defines the operational phase, or for the duration of a login by a user 3 to the database 1. This is illustrated in entry 2 of Figure 5, where the retrieval information is presented without regard to input vector. In this example, the retrieval information that has been compiled in table 18 is the statistic "the rate of retrieving rows was 2000 rows/second across all commands 5". In this example, at step 26, rule 5 from table 6 remains violated, this

time for the operational phase taken as a whole. Thus, at step 28, the entire operational phase is flagged as being suspicious, and the post-flagging protocol 29 performed at step 29 is tailored accordingly.

At step 26, all of the retrieval information in state table 18 can be matched against all of the rules in table 6, or just a subset of the retrieval information and/or a subset of the rules can be used for matching.

An example of an embodiment where table 6 is accessed by two input vectors within the same command 5, as well as by retrieval information from table 18, is illustrated in entries 3 and 4 of Figure 6. Entry 3 gives the rule (rule 6) that for input vector L_1, "no retrievals are allowed between 6 p.m. and midnight unless rule 7 is satisfied". Let us assume that L is the log-in of the user 3 issuing the command 5; L_1 is "Abacus 34"; and retrieval information stored in table 18 for this command 5 specifies that the command 5 was issued at 8 p.m. Then at step 26, computation module 7 determines that rule 6 is violated, unless rule 7 is satisfied. Thus, table 6 must also be accessed by the second input vector, F_1. Let us assume that F is the field being queried by the command 5 and F_1 is the credit card number. Then, computation module 7 looks to table 18 to determine whether the credit card number field is retrieved at a rate D less than 10 per minute by that particular command 5.

The contents of table 6 are generated during an optional training phase, and/or are force fed into table 6 by system administrator 10, and/or are provided by a security or other vendor. A typical training phase is illustrated in Figure 3, and is initiated at step 31. This is done by system administrator 10 flipping a switch (which may be located, for example, on database server 2 or on training module 4); by means of a preselected event occurring (e.g., the first of each month or the addition of a new table within database 1); or by any other means known to one of ordinary skill in the art for starting a computer system.

At step 32, training module 4 observes retrieval commands 5 that users 3 send to database 1. This observation may be done in real time. There are two major ways in which the observing step 32 can be performed: real-time auditing and in-line interception. Real-time auditing is typically used in cases where database 1 has an auditing feature. The auditing information may be placed into an audit table 11 internal to database 1 or into an external database log file 12. In real-time auditing, training module 4 instructs the database 1 to generate a stream of events every time a command 5 enters database 1. The stream can include such items as the text of the command 5, a date/time stamp, information pertaining to the user 3 that issued the command 5, the IP (Internet Protocol) address of the issuing computer 3, the application that issued the command 5, etc.

The stream can appear in string or binary form, and can be extracted using a number of different techniques, depending upon the implementation of the IDS 19, including APIs (Application Programming Interfaces) that access the computer code 1. One example is to use ODBC (Open DataBase Connectivity), a set of C language API's that allows one to examine or modify data within database 1. If the Java programming language is used, JDBC (Java DataBase Connectivity) can be used instead. Another way of extracting the needed information from database 1 is to use code injection or patching to inject logic into one or more modules 1,15 within database server 2, to transfer control to training module 4. In another embodiment, called "direct database integration", the database 1 vendor, who has access to the commands 5 in conjunction with the normal operation of the database 5, makes the commands 5 available to intrusion detection system 19. In yet another embodiment, in cases where database 1 supports it, external database log file 12 may be examined without the need to resort to special software. Once a retrieval command 5 has been processed by training module 4, the command 5 can optionally be expunged from any table or log file it is stored in, to make room for subsequent commands 5.

In in-line interception, at least one of a proxy, firewall, or sniffer 13 is interposed between database 1 and users 3 (see Fig. 1). The proxy, firewall, and/or sniffer 13 examines packets

of information emanating from users 3 and extracts the relevant information therefrom. Proxy, firewall, and/or sniffer 13 may need to decrypt the communications emanating from users 3 if these communications are encrypted.

After a command 5 has been captured in step 32, at step 33 training module 4 observes (extracts) the response of database 1 to the command 5, and updates (augments) state table 18. Step 33 can be performed in real time, i.e., state table 18 can be updated response-by-response. The responses of the database 1 can be extracted using any of the techniques of real-time auditing and/or in-line interception that are described above in conjunction with step 32. Similarly, previously described steps 20 and 22 can be performed using any of the above-described techniques of real-time auditing and/or in-line interception, with computation module 7 rather than training module 4 doing the extraction and generation, respectively.

The operation of step 33 is illustrated in entry 3 of Figure 5. The retrieval information comprises, for the illustrated input vector $L_1F_1A_1$, two retrieval vectors plus statistical information comprising the number of occurrences of each of the retrieval vectors, plus S and D.

Let us assume that L is the parameter "log-in of the user 3 that issued the command 5". The log-in can be some preselected combination of user ID, password, and answer to a challenge

phrase (e.g., "what is your mother's maiden name?"). In this example, L_1 is "Abacus34". F is the field being queried by the command 5. F_1 is "credit card number". A is the application that issued the command 5 or the IP address of the requesting computer 5. A_1 is "Siebel CRM Deluxe Version 22". Let us further assume that during the entirety of the training phase, the only responses generated by database 1 to commands 5 associated with $L_1F_1A_1$ are a plurality of responses having five rows and three columns (retrieval vector 1), and a plurality of responses having seven rows and two columns (retrieval vector 2). Let us further assume that retrieval vector 1 has occurred 963 times, and retrieval vector 2 has occurred 51 times. Thus, the augmentation of state table 18 performed in step 33 for a given command 5 may simply entail incrementing the number of occurrences from 962 to 963, and recalculating S and D. In the illustrated example, the rate S of rows returned by database 1 for this input vector is 1.1 row per second, and the rate D at which database 1 returns columns for this input vector is 2.3 columns per second.

Note that not all of the possible parameters have to be covered in the input vector that is the subject of the training. In this case, just three parameters (out of the many more possible parameters) are so covered (the set of parameters to use may be specified by an administrator 10).

Steps 32 and 33 are repeated for each command 5 that is processed during the training phase.

The training phase is ended, at step 34, by any one of a number of means. For example, system administrator 10 can flip a switch on database server 2 or training module 4. Alternatively, the training phase may end by a statistical technique, e.g., training module 4 monitors the occurrence or frequency of new commonly occurring retrieval vectors. Alternatively, the training phase may end by the occurrence of a preselected elapsed or absolute time, or by any other means known to one of ordinary skill in the art. As with all of the preselected parameters in this patent application, the preselected parameters mentioned in this paragraph may be stored in parameters storage area 17.

At step 35, module 7 converts the retrieval information stored in state table 18 into rules for acceptable and/or unacceptable retrievals within table 6, using preselected set of parameters 17. The administrator 10 may be asked to review and/or augment these rules. Entry 2 of Figure 6 corresponds to entry 3 of Figure 5. There are four rules illustrated for said entry. It can be seen that Rule 1 was derived from the retrieval information in Figure 5 by first concluding that the 963 occurrences of five rows and three columns was greater than a preselected threshold value (e.g., 50) to warrant inclusion in table 6. Then, a preselected margin (in this case, one) in

either direction was applied around the observed numbers of rows and columns to generate the rule. The "AND" following the semicolon in rule 1 is a Boolean AND, i.e., both the criterion "between 4 and 6 rows" and the criterion "between 2 and 4 columns" must be satisfied in order for the retrieval to be deemed acceptable at step 26. There may also be Boolean logic underlying the combination of the rules. For example, in order for module 7 to conclude in step 26 that a retrieval is acceptable, it might have been preselected that either Rule 1 AND Rule 3 AND Rule 4 must be satisfied; OR Rule 2 AND Rule 3 AND Rule 4 must be satisfied in order for the retrieval to be deemed acceptable, where "AND" and "OR" are Boolean operators. If one of these two conditions is not satisfied, module 7 determines that the retrieval is suspicious.

Alternative to a preselected integral margin such as the margin of 1 on either side of the observed numbers of rows and columns illustrated above, any statistical technique may be used to generate the rules of table 6 from the corresponding retrieval information. For example, the margin on the positive side of the number of observations may be a preselected percent of the observed value, or a preselected number of standard deviations. The margin on the lower side of the observed value may be the same or a different percent of the observed value, or the same or a different number of standard deviations. Other statistical

techniques will be readily attainable by those of ordinary skill in the art.

Figure 4 illustrates two optional steps, steps 41 and 42, that constitute the system administrator 10 phase. At step 41, suspicious activity that is observed during the optional training phase is reported to system administrator 10. For example, if the retrieval of a certain combination of rows and columns during the training phase is observed to occur fewer than a preselected threshold number of times, such activity can be flagged to the system administrator 10 as being suspicious. In the above example, suppose that, in addition to five rows and three columns being retrieved 963 times and seven rows and two columns being retrieved 51 times, one row and 100 columns were retrieved one time. This might indicate that the requesting user 3 is attempting to retrieve too much information in a single command 5, and this activity is reported to the system administrator 10 at step 41 as being suspicious.

Similarly, one could incorporate within parameters 17 a maximum number of rows allowed to be retrieved (possibly for a given field/table or set of fields/tables). Let us assume that this maximum number of rows is 20. Then if a particular training phase retrieval attempts to retrieve 21 or more rows, such a retrieval is deemed to be suspicious and is likewise reported to system administrator 10 at step 41. System administrator 10 can

then remove from the set of acceptable retrievals within table 6 such suspicious retrievals.

At step 42, system administrator 10 can force feed rules into table 6. Step 42 can be performed in lieu of or in addition to the training phase. For example, one of the rules provided by the system administrator 10 could be: "no more than 100 rows from CREDIT CARD table are acceptable" or "no more than 100 rows in any one minute from CREDIT CARD table are acceptable".

Rules can also be entirely statistical, such as:

"If the number of rows retrieved by a single user to the CREDIT card field exceeds the historical average for the user's group by more than 2 standard deviations, then generate an alert."

The above description is included to illustrate the operation of the preferred embodiments and is not meant to limit the scope of the invention. The scope of the invention is to be limited only by the following claims. From the above discussion, many variations will be apparent to one skilled in the art that would yet be encompassed by the spirit and scope of the present invention. For example, instead of training the system 19 on the number of columns overall, one could single out certain columns (or combinations of columns) of interest within database 1 and train on that basis, e.g., one could train on the SOCIAL SECURITY

NUMBER column within the PAYROLL table, and/or the CREDIT CARD NUMBER column within the CREDIT INFORMATION table.

What is claimed is: